

## **REMARKS**

Applicant thanks the Examiner for total consideration given the present application. Claims 1, 4, 7-10, 12, 14, 16, 18, 20, and 21 are currently pending, of which claims 1, 9, 16, and 18 are independent. Claims 1, 9, 16, and 18 have been amended through this Reply. Upon careful review, one would conclude that no new matter has been added to application via this amendment. Support for the amendment can be found at least on page 10, paragraph [31], page 11, paragraph [33], page 12, paragraph [35] and paragraph [40] starting on page 15.

Applicant respectfully requests reconsideration of the rejected claims in light of the amendment and remarks presented herein, and earnestly seeks timely allowance of all pending claims.

### **Rejection Under 35 U.S.C. § 103**

The Examiner rejects claims 1, 4, 7-10, 12, 14, 16, 18 and 20-21 under 35 U.S.C. § 103(a) as allegedly being unpatentable over Wall et al. (U.S. Publication No. 2002/0087557)[hereinafter "Wall"] in view of Matula et al. (U.S. Publication No. 2002/0165995)[hereinafter "Matula"] and further in view of Coad et al. U.S. Patent No. 6,851,105)[hereinafter "Coad"]. Applicant respectfully traverses this rejection.

For a Section 103 rejection to be proper, a *prima facie* case of obviousness must be established. *See M.P.E.P. 2142*. One requirement to establish *prima facie* case of obviousness is that the prior art references, when combined, must teach or suggest all claim limitations. *See M.P.E.P. 2142; M.P.E.P. 706.02(j)*. Thus, if the cited references fail to teach or suggest one or more elements, then the rejection is improper and must be withdrawn.

In this instance, in regard to claims 1 and 16, it is respectfully submitted that none of the cited prior art references, alone or in combination, teaches or suggests, *inter alia*,

**"a base filed handler class** which acts as an **intermediary generic mechanism** **when getting the filed values in the model element filed handler object**, wherein the get value function is configured to:

i) dispatch to a sub-classed get field value function in the first subclass which is type safe;

ii) access the model element class and return the attribute value directly upon request; and

iii) provide an entry point for non-type safe application programming interface so that general purpose client code can access the filed values without relying on type safety" (emphasis added.)

In regard to claims 9 and 18, it is respectfully submitted that none of the cited prior art references, alone or in combination, teaches or suggests, *inter alia*,

"a base filed handler class which acts as an intermediary generic mechanism when setting the filed values in the model element filed handler object, wherein the set value function is configured to:

i) provide validation of a new value;

ii) record necessary undo information; and

iii) dispatch to a sub-classed set field value function in the first subclass which is type safe" (emphasis added.)

The claimed invention is directed to efficiently implementing object model attributes. There are several drawbacks in conventional implementation of object model attributes. For example, one drawback of conventional meta-models is that large amounts of memory overhead and computer processing time are required to store, find, and manipulate boxed attribute values within these models. Another drawback of the conventional meta-models is directed to debugging issues. For example, since in the conventional meta-models, values such as the boxed attribute values are typically end up deeply nested within wrapper classes, debugging of generated codes becomes extremely difficult. Further drawback in the conventional meta-models is that complex calling conventions are required to support complicated features or tool, such as undo/redo and notifications which are implemented in many applications.

In order to address the above-mentioned drawbacks, the claimed invention requires, among other features, a base field handler class that acts as an intermediary generic mechanism when setting and getting the field values in the model element object.

The claimed get value function is configured to dispatch to a sub-classed get filed value function in the subclass, which is type safe and directly returns the desired value that is currently stored in the model element. The get value function also provides an entry point for the non-type safe application programming interface so that general purpose client code can access the field values without relying on type safety.

The claimed set value function provides analogous access as get value, but additionally provides validation of a new value, records necessary undo information and dispatches to a sub-classed set field value function in the first subclass which is type safe.

After careful review of the applied prior art references, Applicant respectfully submits that neither the cited portions nor any other portions of Wall, Matula, and Coad teach or suggest the above-identified claim features of claims 1, 9, 16, and 18.

As previously submitted, Wall merely discloses a conventional method and apparatus for providing access control for a decentralized or emergent model on a computer network in which a model is defined using hierarchical relationship of servers, models, and objects.

Matula, on the other hand, discloses a method for dynamic implementation of a Java Metadata Interface (JMI) to a metamodel in which JMI interfaces are implemented as subclasses of handler classes.

Coad is directed to generating, applying, and defining patterns for software development. Although Coad discloses a Singleton pattern in generating a pattern instance, Coad provides no specifics about any base filed handler class which acts as an intermediary generic mechanism when setting and getting the filed values in the model element filed handler object nor does Coad teach or suggest the above-identified claimed configuration of the “get value function” and the “set value function”.

Therefore, for at least these reasons, it is respectfully submitted that independent claims 1, 9, 16, and 18 are allowable over Wall, Matula, and Coad. Claims 4, 7, 8, 10, 12, 14, 20, and 21 are at least allowable by virtue of their dependency on corresponding allowable independent claim.

**CONCLUSION**

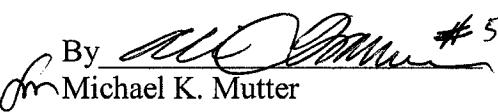
In view of the above amendment, Applicant believes the pending application is in condition for allowance.

Should there be any outstanding matters that need to be resolved in the present application, the Examiner is respectfully requested to contact Ali M. Imam Reg. No. 58,755 at the telephone number of the undersigned below, to conduct an interview in an effort to expedite prosecution in connection with the present application.

If necessary, the Commissioner is hereby authorized in this, concurrent, and future replies to charge payment or credit any overpayment to Deposit Account No. 02-2448 for any additional fees required under 37.C.F.R. §§1.16 or 1.147; particularly, extension of time fees.

Dated: October 9, 2008

Respectfully submitted,

By   
Michael K. Mutter # 58755  
Registration No.: 29,680  
BIRCH, STEWART, KOLASCH & BIRCH, LLP  
8110 Gatehouse Road  
Suite 100 East  
P.O. Box 747  
Falls Church, Virginia 22040-0747  
(703) 205-8000  
Attorney for Applicant